CSC3100 Data Structures Spring 2022

Assignment 2

Due: 23:59, Mar 20, 2022

1. Problems

1.1. Problem 1: Two Sum

Description

Given an array of n integers and an integer t, find the indices of the two elements such that they add up to t. The indices of elements in the array are from 1, 2, ... up to n. Each input would have **exactly one** solution. You cannot use the same element twice. You need to output the two indices in order from smallest to largest.

Input

- Line 1: two integers n, t.
- Line 2: n integers a_1, a_2, ..., a_n, representing the integers in the array.

Output

- Two integers separated by spaces, representing the two indices in order from smallest to largest.

Limits

- $2 \leq n \leq 5 \times 10^5$.
- $-10^9 \leq t \leq 10^9$.
- $-10^9 \leq a_i \leq 10^9$.

Note

- For 20% data, $n \leq 5$.
- For 80% data, $n \leq 5000$.
- For 100% data, $n \leq 500000$.

Sample I

Input

```
4 9
2 7 11 15
```

Output

```
1 2
```

Explanation

Since $a_1 + a_2 = 9 = t$, we output `1 2`.

Sample II

Input

```
3 6
3 2 4
```

Output

```
2 3
```

Sample III

Input

```
2 6
3 3
```

Output

```
1 2
```

1.2. Problem 2: Inversion Number

Description

Let a be an array. If $i < j$ and $a_j < a_i$, the unordered pair of indices (i, j) is called an inversion of a. Now given an array a of n integers, count the number of inversions of the array a. The elements in the array a may be the same.

Input

- Line 1: one integer n.
- Line 2: n integers a_1, a_2, ..., a_n, representing the integers in the array.

Output

- One integer representing the number of inversions of the array a.

Limits

- $1 \leq n \leq 5 \times 10^5$.
- $-10^9 \leq a_i \leq 10^9$.

Note

- For 20% data, $n \leq 5$.
- For 80% data, $n \leq 5000$.
- For 100% data, $n \leq 500000$.

Sample I

Input

```
4
5 4 2 6
```

Output

```
3
```

Explanation

The inversions of the array are $(5, 4)$, $(5, 2)$, $(4, 2)$, then we output `3`.

Sample II

Input

```
6
5 4 2 6 3 1
```

Output

```
11
```

Sample III

Input

```
6
1 1 2 4 3 2
```

Output

```
3
```

1.3. Problem 3: Line Arrangement

Description

A teacher in a school wants to arrange the class of n students in a line, and the students are numbered from 1 to n.

He takes the following approach:

1. The student numbered 1 is placed into the line. Now he is the only one in the line.
2. The students numbered $2 \sim n$ are placed in the line in turn. The teacher designates the student numbered x to stand on the left or the right of one of the students numbered $1 \sim x-1$ (i.e., a student who have been previously placed in the line).
3. Remove m students from the line and remain the order of the other students' positions unchanged.

After all the students have been arranged in the line as described above, the teacher wants to know the numbers of all the students from left to right.

Input

- Line 1: one integer n, representing the number of the students.
- Line 2 to n: two integers x_i, p_i where i is the line number, representing how students will be placed into the line. If $p_i = 0$, the student numbered x_i will stand on the left of the student numbered i. If $p_i = 1$, the student numbered x_i will stand on the right of the student numbered i.
- Line $n+1$: one integer m, representing the number of the students to be removed.
- Line $n+2$: m integers y_1, y_2, ..., y_m, representing which students will be be removed from the line.

Output

- n integers separated by spaces, representing the number of all students in the line from left to right.

Limits

- $2 \leq n \leq 5 \times 10^5$.
- $1 \leq x_i < i$ where i is the line number.
- $p_i \in \{0, 1\}$.
- $1 \leq m < n$.
- $1 \leq y_i \leq n$; $y_i \neq y_j, \forall i \neq j$.

Note

- For 20% data, $n \leq 5$.
- For 80% data, $n \leq 5000$.
- For 100% data, $n \leq 500000$.

Sample I

Input

```
4
1 0
2 1
1 0
2
3 4
```

Output

```
2 1
```

Explanation

- The student numbered 1 is placed into the line. Then the line is $[1]$.
- The student numbered 2 is placed into the line. He/she stands on the left of the student numbered 1. Then the line is $[2, 1]$.
- The student numbered 3 is placed into the line. He/she stands on the right of the student numbered 2. Then the line is $[2, 3, 1]$.
- The student numbered 4 is placed into the line. He/she stands on the left of the student numbered 1. Then the line is $[2, 3, 4, 1]$.
- The student numbered 3 is removed from the line. Then the line is $[2, 4, 1]$.
- The student numbered 4 is removed from the line. Then the line is $[2, 1]$.

Sample II

Input

```
5
1 0
1 1
3 1
4 0
1
2
```

Output

```
1 3 5 4
```

Sample III

Input

```
5
1 0
1 0
1 0
1 0
4
1 2 3 4
```

Output

```
5
```

2. Requirements

You can write your code in Java, Python, C, or C++. The time limit may vary among different languages, depending on the performance of the languages. Your code must be a complete runnable program instead of only a function. We guarantee that the input data used to test strictly conforms to the requirements in the description, and you do not need to deal with cases where the input data is invalid. If you do not quite understand, you can see the following example.

You also need to write a report to explain the following:

- What are the possible solutions for this problem?
- How do you solve this problem?
- Why is your solution better than others?

Please note that the maximum number of pages allowed for your report is **10 pages**.

3. Example

3.1. Example Problem: A + *B*

Description

Given two integers A and B, compute and print $A + B$.

Input

- Line 1: two integers A, B.

Output

- One integer $A + B$.

Limits

- $0 \leq A, B \leq 10^6$.

Sample

Input

```
1 2
```

Output

```
3
```

Explanation

Since $1 + 2 = 3$, we output `3`.

3.2. Solutions

Java

```java
import java.util.Scanner;

public class ExampleProblem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int a = scanner.nextInt();
        int b = scanner.nextInt();
        System.out.println(a + b);
        scanner.close();
    }
}
```

Python

```python
line = input()
tokens = line.split(' ')
a = int(tokens[0])
b = int(tokens[1])
print(a + b)
```

C

```c
#include <stdio.h>

int main(void)
{
    int a, b;
    scanf("%d%d", a, b);
    print("%d\n", a + b);
    return 0;
}
```

C++

```cpp
#include <iostream>

int main()
{
    int a, b;
    std::cin >> a >> b;
    std::cout << a + b << std::endl;
    return 0;
}
```

4. Submission

4.1. Online Judge

Preparation

Our Online Judge platform for this course is located at `https://106.13.15.234`. You can also visit via `https://cuhkszoj.com`.

When you visit the website for the first time, it will warn you that the website is not secure. At this point, please just ignore it. Our website will not steal your personal information.

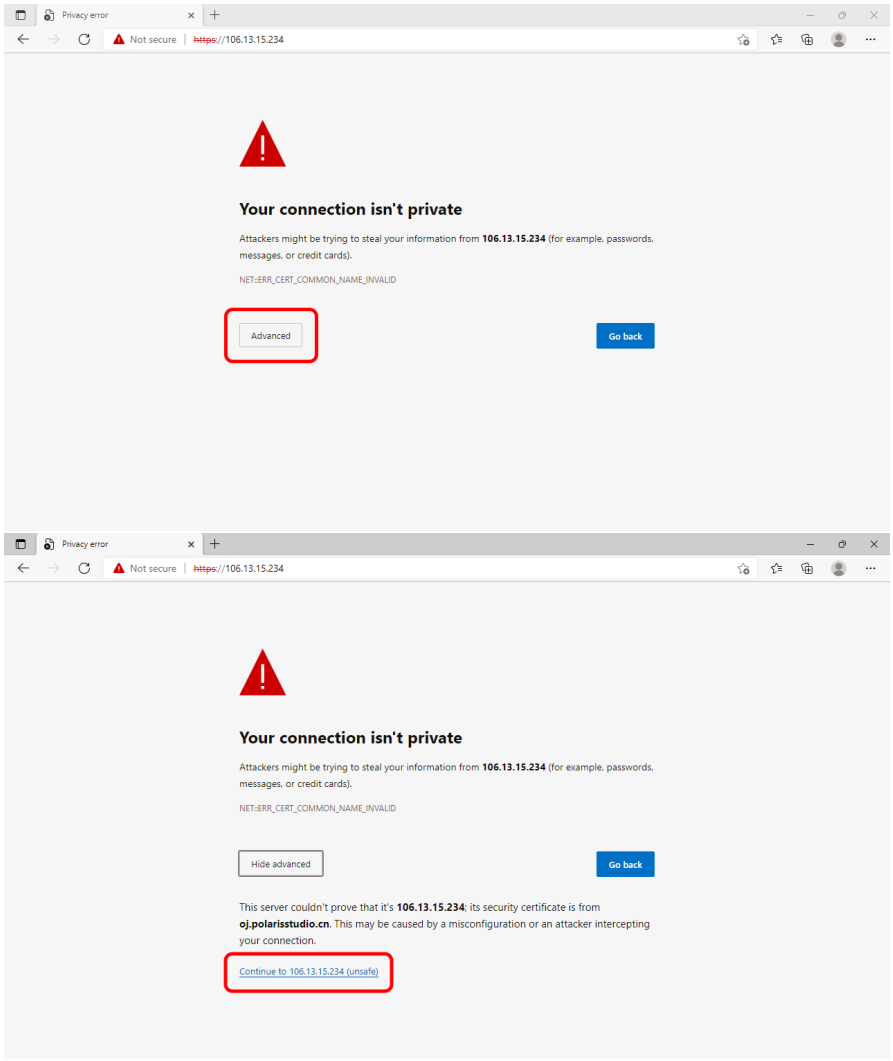

Figure 1: Steps to ignore security warnings

Registration

You can register an account on the website with any email, but **your username must be your student ID**. If you already have an account on this website and your username is not your student ID, you can choose to register an account with another email address, or you can

contact the administrator (HU Haichuan, 119010103@link.cuhk.edu.cn) to change your name.

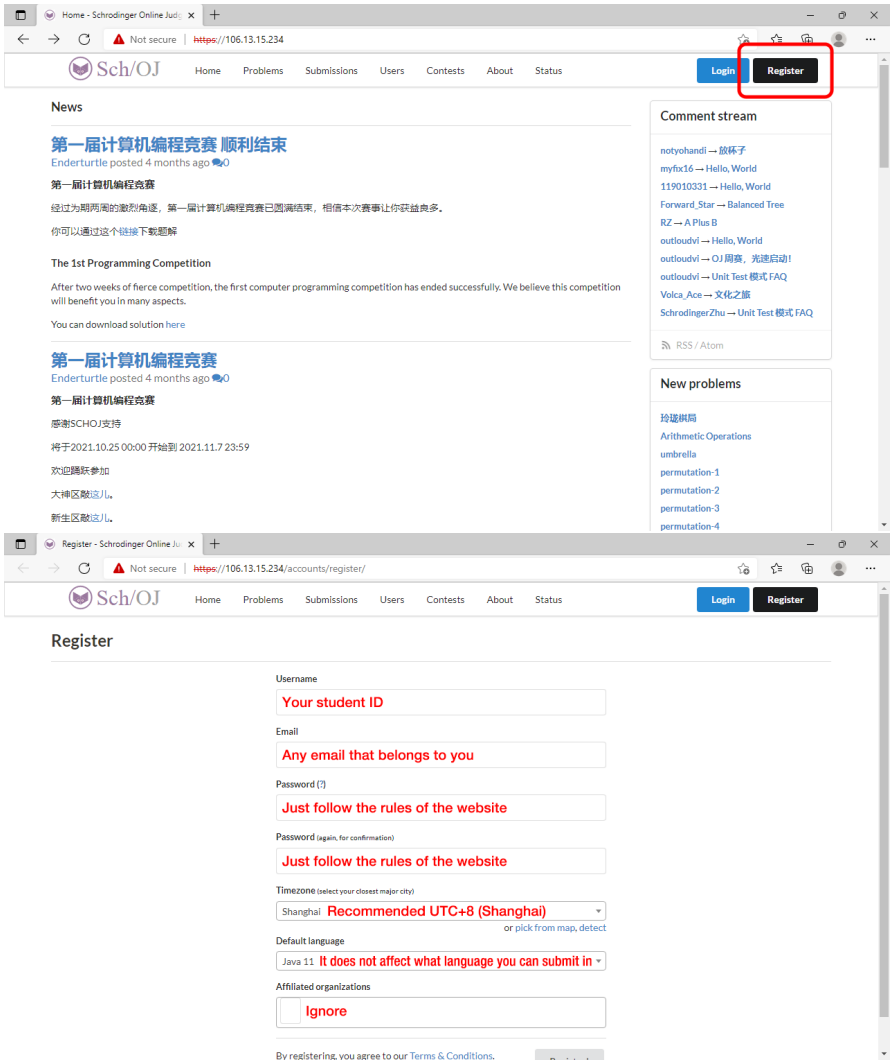

Figure 2: Steps to register

Join

We have placed this assignment in the "Contests" board under the name "CSC3100 Spring 2022 Assignment 1". You can also visit the page of this assignment directly through https://106.13.15.234/contest/csc3100spr2022a2. Then you can join this "contest".

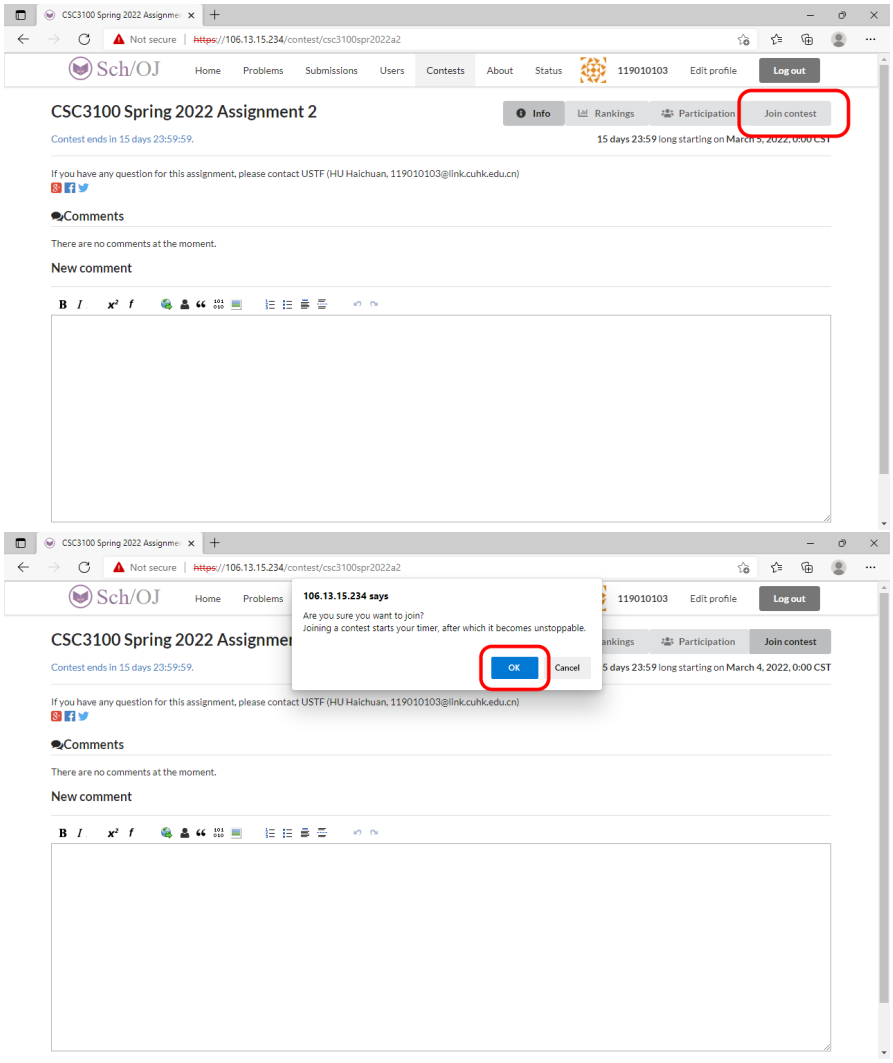

Figure 3: Steps to join the "contest"

Submission

Once you have completed one problem, you can submit your code on the corresponding page on the Online Judge platform in order to get grades for the code part. You have multiple chances to submit, and we will choose your highest one for grading. Each problem has multiple test points of different difficulty. You will get a part of the score even if your algorithm is not the best.

4.2. Blackboard

You also need to upload your **code and report** to the Blackboard platform. You need to name each of your files according to the following rules and compress them into a `zip` archive, where *ID* is your student ID.

```
A2_Submission_ID.zip
├── A2_P1_ID.java/py/c/cpp/cc
├── A2_P2_ID.java/py/c/cpp/cc
├── A2_P3_ID.java/py/c/cpp/cc
└── A2_Report_ID.pdf
```

5. Grading

- Code (80%): strictly dependent on your score on the Online Judge platform
  - Problem 1 (25%), Problem 2 (25%), Problem 3 (30%)
- Report (20%)

6. Note

If you have any question for this assignment, please contact HU Haichuan (119010103@link.cuhk.edu.cn).